

ESP Normalization Exercise Suite

Customer Details and Customer Orders

Customer Details View

Customer Details	Customer Number: 137
Name:	Fred Smith
Address:	123 SomeWhere St.
City:	Edmonton
Province:	Alberta
Postal Code:	T5H 2J9
Home Phone:	436 - 7867

ONF: Discover View Data Attributes

The process of normalization starts by pooling the known data. Data has 2 components; an identifier (what the data represents) and the value. The data identifier is a description of the piece of information. This data can be obtained from several sources such as reports, company documents, employee knowledge, etc. Here we are given the customer card.

The first step of ONF is to identify the data from the Customer Details View source document. The customer card has many pieces of data all identified by a label; which can be used as the data identifiers. The customer card has a Customer Number label. In our example, this piece of information has a data value of 137. The value could have been 111 or 2009 or some other numerical value. The value is unimportant at this time; it is the label that describes what the value stands for that is important.

The first step is to collect and record the descriptive labels which will be used identify a piece of data.

Customer(CustomerNumber, Name, Address, City, Province, PostalCode, HomePhone)

Once the labels (also known as data attributes) are recorded, they can be further examined for appropriateness.

Atomization

Atomizing a data attribute reduces any attribute to its smallest components without the attribute losing meaning. Name can be reduced to a first name and a last name. All other attributes are at their atomized state.

Atomize any fields ie: Name becomes FirstName and LastName

Repeating Groups

A repeating group attribute is an attribute that has two or more possible values on the same document. An example can be seen on the second source document of the exercise, Customer Order View, where there are 3 items description (each line represents an item). The column headers represent the data attribute. Check for any repeating groups on the Customer Detail View. The customer card has no repeating groups.

Pick a primary key.

Picking a primary key requires identifying one or more data attributes to be the primary key for your data attribute collection. This attribute(s) should have a unique value which can be used as a reference to obtain a single record of data values. The attribute(s) which meet this criterion are called candidate keys. Candidate key(s) that appear on your document are called natural keys. Sometimes no attribute can be found to meet this criterion; in which case; you create another data attribute that will have the sole purpose of being the record's unique value field. This key is called a technique key.

Often, your documents will come with business rules. The following rules were stated in our example:

- Customer numbers are unique and are never reused by the company.
- A card is created for each new customer that buys from the company.

For our customer collection, the customer number is unique. This data attribute will become our primary key.

Customer(**CustomerNumber**, FirstName, LastName, Address, City, Province, PostalCode, HomePhone)

Finally, one should identify this group of attributes with a descriptive name; Customer. This collection of data attributes is referred to as an entity. An entity that has repeating groups is said to be in 0 Normal Form.

1NF: Remove Repeating Groups

An entity in 1NF (first normal form) does not contain any repeating groups. The Customer entity does not contain any repeating groups. The Customer entity is in first normal form.

2NF: Check for Partial Dependencies

An entity in 2NF (second normal form) does not contain any partial dependencies. An entity must have a compound primary key (2 or more attributes which are needed to uniquely identify a single record) for a partial dependency to exist. There is no compound primary key in the Customer entity; therefore, the entity is in 2NF.

3NF: Check for Transitive Dependencies.

An entity in 3NF (third normal form) does not contain any transitive dependencies. An entity must have a 2 or more non primary key attributes for a partial dependency to exist. A transitive dependency occurs when a pair of non primary key attributes is closely associated with each other such that they can form their own entity. The first name and last name attributes could be considered a transitive dependency; Name. However, does a person's first name really depend on their last name? Our Customer entity does not have any transitive dependencies; therefore is in 3NF.

Final set of 3NF entities

The Customer Detail View has produced a single entity from the normalization process.

Customer(**CustomerNumber**, FirstName, LastName, Address, City, Province, PostalCode, HomePhone)

Customer Order View



Customer Name: Fred Smith
Address: 123 SomeWhere St.
Edmonton, Ab T5H 2J9
Phone: 436 - 7867

Customer Number: 137

Date: Jan 16, 2015

Order Number: 219

Item Number	Description	Quantity	Price	Amount
H23	Heater Fan Belt - 23"	1	11.99	11.99
H319	Heater Fan Belt Support Brackets	2	4.99	9.98
M24	Bolts - 24 mm	8	0.29	2.32
			Subtotal	24.29
			:	
			GST:	1.70
			Total:	25.99

Business Rules:

- Each order form has a preprinted sequential order number that is unique.
- The current price for each item in inventory.
- The price each item sold for at the time the item was sold.
- Item Number and Customer Numbers are unique within the business.

0NF: Discover View Data Attributes

Let us start the normalization process by recording all the data attributes. Remember a data attribute describes the data; it is not the data value. Create the data attributes by using the labels beside each piece of data. Use the column headers as the attribute names for the item data. The order of the data attributes is not important.

(CustomerNumber, CustomerName, Address, Phone, Date, OrderNumber, ItemNumber, Description, Quantity, Price, Amount, Subtotal, GST, Total)

According to a business rule, a new attribute must be introduced to handle the historical nature of Price. Price can change over time. Imagine selling an item in May; then the price changes in July. Over the year, we sold the same item for 2 different prices. This situation is typically resolved by having 2 attributes. One attribute will contain the current selling price while the other the selling price at the time the order was created. We will call the current selling price attribute CurrentPrice. The second attribute, price at the time the order was placed, will be called SellingPrice.

(CustomerNumber, CustomerName, Address, Phone, Date, OrderNumber, ItemNumber, Description, Quantity, CurrentPrice, SellingPrice, Amount, Subtotal, GST, Total)

If you do not discover the CurrentPrice/SellingPrice attributes until later in your normalization process, then, you should start from ONF again with the new attributes in the original list.

Atomization

Customer Name and Address can be atomized into small data attributes.

(CustomerNumber, **FirstName, LastName, Address, City, Province, PostalCode**, Phone, Date, OrderNumber, ItemNumber, Description, Quantity, CurrentPrice, SellingPrice, Amount, Subtotal, GST, Total)

Repeating Groups

A repeating group attribute is an attribute that has two or more possible values on the same document. An example can be seen on the Customer Order where there are 3 items description (each line represents an item). We used the column headers represent the repeating group data attributes. Group these attributes in the list by placing parenthesis around the attributes. This grouping is a repeating group.

(CustomerNumber, FirstName, LastName, Address, City, Province, PostalCode, Phone, Date, OrderNumber, **(ItemNumber, Description, Quantity, CurrentPrice, SellingPrice, Amount)**, Subtotal, GST, Total)

Pick a primary key.

This customer order view has 3 attributes which are unique within the business; order number, item number and customer number. Each of the candidate keys are natural keys (appearing on the document). The task is to pick the least number of data attributes to uniquely identify a particular instance of the customer order.

Item number, though unique in identifying an item, is not a singular value on this document. Therefore, we could discount this attribute as a candidate, by itself, to be the best primary key.

Both customer number and order number are singular values on this document. How do you choose between the two? Examine the document. Is the document about the customer or about the order? There is customer data on the document. However, the document is more than just the customer data; it is about an order for the listed customer. So, just like an election, the order number encompasses more of the data than just the customer number and thus should be picked as the primary key.

Order(CustomerNumber, FirstName, LastName, Address, City, Province, PostalCode, Phone, Date, **OrderNumber**, (ItemNumber, Description, Quantity, CurrentPrice, SellingPrice, Amount), Subtotal, GST, Total)

Finally, we will reference this collection by the name of Order. Since the entity contains a repeating group, it is said to be in 0NF (zero normal form).

1NF: Remove Repeating Groups

Entities in 1NF must not have repeating groups. The 0NF entity has a repeating group identifying an item on the order. One can recognize the repeating group by having a data attribute with two or more values associated with it on the document. On the order entity: ItemNumber, Description, Quantity, CurrentPrice, SellingPrice, Amount are the repeating attributes. These attributes are removed from the Order entity and placed in a second entity.

Removing repeating groups:

- a) create a new entity containing the repeating group and the primary key (OrderNumber) from the 0NF entity
- b) Make OrderNumber (the primary key from the 0NF entity) in your new entity a Foreign Key.
- c) Choose one or more attributes in the new entity to create the primary key. It may be necessary to combine OrderNumber with another attribute for form the primary key for the new entity. This will be the compound primary key for the new entity.
- d) Give your new entity a name: OrderDetail

Order(CustomerNumber, FirstName, LastName, Address, City, Province, PostalCode, Phone, Date, **OrderNumber**, Subtotal, GST, Total)

OrderDetail(**OrderNumber**, **ItemNumber**, Description, Quantity, CurrentPrice, SellingPrice, Amount)

2NF: Check for Partial Dependencies

You must have a compound primary key in the entity table for any partial dependency to exist. The OrderDetail has a compound primary key. This entity must be checked for partial dependencies. A partial dependency is any non primary attribute closely associated with only part of the compound primary key.

Attribute: Description

This attribute can be associated (dependent) very closely with the ItemNumber. It has no dependence on the OrderNumber. View another way, the description describes the item, regardless of which order(s) have the item on it. Description is not dependent on the order, just the item. Description is said to be partially dependent on the primary key (OrderNumber/ItemNumber).

Attribute: CurrentPrice

CurrentPrice depends on (very closely associated/tied to) the ItemNumber. It is the price of the item at this point in time.

Attribute: SellingPrice, Quantity and Amount

Selling, Quantity and Amount depends on both the item and the order. Said another way, for a particular order at some point in time, I paid a price for a certain quantity totaling to a specific amount. Another order at another time could have the same data values or different values. These attributes are wholly dependent on both parts of the compound primary key, I must know which order, which item. These two attributes will remain with the OrderDetail entity.

Removing Partial Dependencies

- Create a new entity containing a copy of the partial primary key from the 1NF entity and make it the primary key of the new entity.
- Make the partial primary key from the “sending” entity a Foreign Key.
- Remove the dependent non primary key attribute from the “sending” entity to your new entity.

OrderDetail(**OrderNumber**, **ItemNumber**, Quantity, SellingPrice, Amount)

Item(**ItemNumber**, Description, CurrentPrice)

.

Final set of 2NF entities.

Order(CustomerNumber, FirstName, LastName, Address, City, Province, PostalCode, Phone, Date, **OrderNumber**, Subtotal, GST, Total)

OrderDetail(**OrderNumber**, **ItemNumber**, Quantity, SellingPrice, Amount)

Item(**ItemNumber**, Description, CurrentPrice)

3NF: Check for Transitive Dependencies.

You must have a multiple non-primary key attributes in the entity for any transitive dependency to exist. A transitive dependency is the close association of two non primary key attributes such that these attributes can form a separate entity.

Order(CustomerNumber, FirstName, LastName, Address, City, Province, PostalCode, Phone, Date, **OrderNumber**, GST, OrderTotal)

In Order, the FirstName, LastName, Address, City, Province, PostalCode and Phone depend on (very closely associated/tied to) the CustomerNumber. These attributes can be removed to a separate entity. The only attribute that is really required to be kept on the Order entity is the customer number. FirstName, LastName, Address, City, Province, PostalCode and Phone are considered to be transitive dependent attributes for CustomerNumber.

Steps:

- a) Create a new entity containing all attributes of the transitive relationship(s).
- b) Remove the attributes from the “sending” entity.
- c) Make a primary key from these attributes in your new entity.
- d) Place a copy of this attribute back in the “sending” entity (Order) and make it a foreign key.

Order(*CustomerNumber*, Date, **OrderNumber**, Subtotal, GST, Total)

Customer(**CustomerNumber**, FirstName, LastName, Address, City, Province, PostalCode, Phone)

Examining the OrderDetail and Item table, we find there is no close association between their non primary key attributes. Quantity is not dependent on SellingPrice nor is SellingPrice dependent on Amount. Description is not dependent on CurrentPrice. There are no transitive dependencies in these entities.

Final set of 3NF entities

Order(*CustomerNumber*, Date, **OrderNumber**, Subtotal, GST, Total)

Customer(**CustomerNumber**, FirstName, LastName, Address, City, Province, PostalCode, Phone)

OrderDetail(**OrderNumber**, **ItemNumber**, Quantity, SellingPrice, Amount)

Item(**ItemNumber**, Description, CurrentPrice)

Merging Views.

The final step in our solution is to merge the two views above to make one set of entities. You can do this by looking for entities that have the same (identical) primary keys.

Steps:

- a) Create one entity with the existing primary key and all non-primary attributes.
- b) Remove any duplicate attributes so each attribute appears only once. You may find that common attributes have different

names. You must choose one name to represent the attribute. Example Phone and HomePhone. These are the same attribute. We will choose HomePhone because it is more descriptive.

Customer(**CustomerNumber**, FirstName, LastName, Address, City, Province, PostalCode, HomePhone)

and

Customer(**CustomerNumber**, FirstName, LastName, Address, City, Province, PostalCode, Phone)

Gives

Customer(**CustomerNumber**, FirstName, LastName, Address, City, Province, PostalCode, HomePhone)

Finally set of entities

Order(**OrderNumber**, *CustomerNumber*, Date, Subtotal, GST, Total)

OrderDetail(**OrderNumber**, **ItemNumber**, Quantity, SellingPrice, Amount)

Item(**ItemNumber**, Description, CurrentPrice)

Customer(**CustomerNumber**, FirstName, LastName, Address, City, Province, PostalCode, HomePhone)

Derived Attributes

One final note about derived attributes. Depending on your “shop” standards, derived attributes may be removed from all entities since they could be calculated at a later time. Subtotal, Total and Amount could all be recalculated at a later time. These attributes could be removed from the solution. GST is a calculated field but is really dependent on the current GST rate. This attribute could be removed but would have to be replaced with a new attribute reflecting the GST rate at the time of the order.